

# Sistem automatizat de udare a plantelor

- Toader Mihai-Cristian
- 332CD
- Indrumator: Razvan Virtan

## Introducere

Proiectul constă în realizarea unui sistem automatizat de udare a plantelor controlat prin intermediul unui microcontroler. Sistemul utilizează senzori de umiditate a solului și temperatura pentru a determina când o plantă necesită apă și o pompă de apă pentru a livra cantitatea necesară de apă direct în sol.

Sistemul este conceput să fie simplu fiind potrivit pentru plante de interior sau pentru micile amenajări de grădină. Utilizatorii pot uda planta manual folosind o interfață simplă și pot citi datele precum umiditatea și temperatura.

## Descriere Generală

Utilizatorii vor interacționa cu sistemul de udare a plantelor printr-o interfață de bază prin care vor putea citi factorii de mediu din jurul plantei, cum ar fi temperatura, lumina și umiditatea solului. Acestea vor fi primite prin Wifi de la plăcuță după ce au fost măsurate de senzori. Când nivelul de umiditate este prea scăzut, sistemul va iriga automat planta după ce verifică cantitatea de apă din rezervor. În cazul lipsei de apă din rezervor utilizatorul va primi un mesaj prin aplicație.



## Designul Hardware

În realizarea proiectului voi folosi:

- Arduino UNO
- Senzor umiditate sol
- Senzor temperatura și umiditate

-Fotorezistor

-Senzor ultrasonic

-Pompa de apa

-Modul Releu



Mai jos am o schema orientativa a pieselor hardware, va fi inlocuita odata ce ajung ultimele piese si sistemul este pus in functiune.



## Designul Software

Descrierea codului aplicației (firmware):

- Mediu de dezvoltare utilizat (de exemplu, Arduino IDE)
- Biblioteci și surse terțe parte utilizate (de exemplu, biblioteci de senzori Adafruit)
- Algoritmi și structuri pe care plănuți să le implementați
- (Etapă 3) Surse și funcții implementate

### Biblioteci folosite:

ESP8266WiFi.h: Este folosită pentru gestionarea conexiunilor WiFi și a comunicațiilor de rețea pentru placa folosită.

BlynkSimpleEsp8266.h: Biblioteca Blynk permite interacțiunea cu platforma Blynk, care este folosită pentru a monitoriza și controla sistemul în timp real printr-o aplicație mobilă

DHT.h: Utilizată pentru a interfața cu senzorii de temperatură și umiditate DHT11 sau DHT22.

NewPing.h: Utilizată pentru a interfața cu senzorul ultrasonic HC-SR04.

### Schelet cod:

1. Conectarea la WiFi:

- NodeMCU se conectează la rețeaua WiFi utilizând credențialele furnizate.
- Biblioteca ESP8266WiFi.h gestionează conexiunea la rețea.

2. Monitorizarea umidității solului:

- Senzorul de umiditate a solului este citit periodic.

- Datele sunt procesate și trimise către Blynk pentru monitorizare în timp real.

### 3. Controlul pompei de apă:

- Pompa de apă este controlată pe baza valorilor citite de la senzorul de umiditate a solului și de la senzorul ultrasonic.
- Biblioteca NewPing.h este folosită pentru măsurarea distanței de la senzorul ultrasonic.

### 4. Monitorizarea temperaturii și umidității:

- Senzorul DHT11 măsoară temperatura și umiditatea ambientale.
- Biblioteca DHT.h este folosită pentru citirea datelor de la senzorul DHT11.

### 5. Monitorizarea luminozității:

- Un LDR (Light Dependent Resistor) măsoară intensitatea luminii.
- Valorile sunt prelucrate și trimise către Blynk pentru analiză.

### 6. Monitorizarea nivelului apei în rezervor:

- Senzorul ultrasonic măsoară distanța până la suprafața apei pentru a detecta nivelul apei.
- Dacă nivelul apei este scăzut, se trimite o notificare prin Blynk.

### **Testarea si Calibrarea senzorilor:**

Senzorul de temperatura: a fost testat si observat ca da valori normale, corecte in relatie cu temperatura de afara

Senzorul ultrasonic: calibrarea a fost realizată prin compararea unor distante cunoscute pentru valida corectitudinea citirilor.

Fotorezistor: L-am pus sub diferite nivele de lumina, de la intunericul palmei pana la flashul telefonului.

## **Concluzii**

Proiectul a fost o provocare, de la calibrarea senzorilor pana la utilizarea platformei Blynk si sincronizarea acesteia. Pe partea fizica, a fost destul de dificil sa gasesc un mod de organizare a pieselor pentru a imi oferi un mod usor de lucru. Cand vine vorba de cod, folosirea functionalitatilor blynk a fost destul de friendly, dar a trebuit sa ma interesez de functionarea fiecarui senzor in parte.

## **Descărcare**

O arhivă (sau mai multe, dacă este cazul) conținând toate fișierele produse în timpul proiectului: coduri sursă, scheme, etc. Includerea unui fișier README, un ChangeLog și un script de compilare și încărcare automată pe microcontroler face întotdeauna o impresie bună.

Fișierele se încarcă pe wiki folosind funcția "Add Images or other files". Spațiul de nume pentru încărcarea fișierelor ar trebui să fie de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume\_student** (dacă este cazul). **Exemplu:** Ionescu Maria, 331CC → **:pm:prj2024:cc:ionescu\_maria**.

## Jurnal

Puteți avea și o secțiune de jurnal în care asistentul de proiect poate urmări progresul proiectului.

## Bibliografie/Resurse

Listă cu documente, fișe tehnice, resurse Internet utilizate, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

Cod: <https://github.com/Ancuu/Sistem-Irigare-Blynk>

Proiect similar, mai simplu:

<https://srituhobby.com/how-to-make-a-plant-watering-system-with-the-nodemcu-esp8266-board-and-the-new-blynk-update/>

Blynk: <https://www.blynk.io/>

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2024/rvirtan/smart\\_plant\\_irrigation](http://ocw.cs.pub.ro/courses/pm/prj2024/rvirtan/smart_plant_irrigation)



Last update: **2024/05/26 10:44**